REMARKS

Claims 1-23 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks. Applicants note that the Disposition of Claims noted in the Office Action Summary is incorrect. Claims 7, 14 and 18 are listed as being rejected, but they are not. These claims are objected to as being dependent on a rejected base claim, but otherwise allowable.

Section 103(a) Rejections:

The Examiner rejected claims 1-5, 8-12, 15 and 19-23 under 35 U.S.C. § 103(a) as being unpatentable over Jourdan (U.S. Patent 6,848,031) in view of Darcy (U.S. Patent 7,325,097), and claims 6, 13, 16 and 17 as being unpatentable over Jourdan and Darcy in view of prior art. Applicants traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, the cited art fails to teach or suggest *wherein the prefetch unit is further configured to prefetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache*. The Examiner notes that Jourdan discloses a first level trace cache, that, when a miss occurs, a second level cache is accessed, and that Jourdan discloses a method of accessing the memory when a cache miss occurs. The Examiner admits that Jordan fails to teach the above-referenced limitation and relies on Darcy to teach it. The Examiner submits, "Darcy discloses a demotion system, allowing a higher level cache to demote a block of instructions to a lower level cache in response to an eviction," citing column 30, lines 6-25.

The Examiner has mischaracterized the teachings of Darcy. The cited passage actually states, in part, "when a block is replaced from the top cache in the stack (again using any suitable replacement algorithm, such as an LRU algorithm), the block is demoted to the next lowest level in the cache. This can cause a domino effect wherein each cache in the stack inserts a block into the cache below it in the stack, causing the

lower cache to replace one of its blocks, so that blocks are replaced from one level in the stack to the next, until a block is pushed out of the lowest cache in the stack" (emphasis added). In other words, in Darcy, when a block is demoted from a higher level cache, it is pushed to a lower level cache by the higher level cache. This clearly does not involve a prefetch unit prefetching a line of instructions into the lower level cache, according to the limitations of Applicants' claim, "wherein the prefetch unit is configured to prefetch instruction code **from a system memory** for storage within the instruction cache." Therefore, even if the system of Jourdan included the demotion feature of Darcy, the combination would not teach or suggest the above-referenced limitation of claim 1. Applicants note that Darcy goes on to describe, in column 30, lines 26-29, "In accordance with one embodiment of the present invention, blocks replaced out of the lowest cache in the stack are temporarily stored in a resource referred to herein as a victim repository." Therefore, even when a block is evicted from the lowest level cache of Darcy, the recited response, "*the prefetch unit is further configured to prefetch a line of instructions into the instruction cache"* is not carried out. **In light of the discussion above, Applicants assert that not only does Darcy not teach prefetching a line of instructions into the instruction cache in response to a higher level cache eviction, Darcy's demotion mechanism teaches away from any need for such a prefetch.**

In addition, Applicants assert that it is not clear that the demotion mechanism of Darcy could even be applied to the trace cache and instruction cache of Jourdan, as suggested by the Examiner. Jourdan (in column 1 lines 34-46) describes, "For example, the first level might be a trace cache 26, which stores micro-operations (pops) in traces that correspond to various paths through the logic of the program code... The trace cache 26 is typically located between the decoder 24 and the execution core 27, and can generally be accessed at full processor speed. On the other hand, the second level of the cache system, such as the instruction cache 22, is often located between the decoder 24 and the remainder of the front end, which includes main memory 29" (emphasis added). In other words, as illustrated in FIG. 6, the trace cache of Jourdan stores decoded instructions, while the instruction cache of Jourdan stores undecoded instructions. Therefore, it is not clear if, or how, a demotion mechanism as

taught by Darcy could be employed to push traces of micro-operations into an instruction cache that is configured to store undecoded instructions and pass them to a decoder (as shown in FIG. 6). **A modification regarding the type or form of instructions or micro-operations stored in Jourdan's two caches to support the demotion mechanism of Darcy would change the principle of operation of those caches.**

Applicants remind the Examiner that "It is improper to combine references where the references teach away from their combination." *In re Grasselli*, 218 USPQ 769, 779 (Fed. Cir. 1983). In addition, "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Accordingly, Applicants assert that the Examiner has failed to establish *prima facie* obviousness.

The Examiner submits, "Jourdan would have been motivated to utilize this technique in order to prevent an eviction from requiring a slow direct memory access," and "It would have been obvious at the time of the invention for one of ordinary skill in the art to take the processing system of Jourdan and allow an eviction to demote a block of instructions as in Darcy. The combination would result in the instruction cache prefetching a line of instructions into the instruction cache (from a demotion) in response to an eviction from the trace cache (the higher level cache). This line of instructions is clearly not currently needed for execution, but prefetched in anticipation of a re-execution of these instructions." **However, as discussed above, Darcy does not teach or suggest prefetching instructions (from system memory) in response to a cache eviction of any kind, much less a trace cache eviction, and instead teaches away from any need for such a prefetch.** Therefore, contrary to the Examiner's assertion, the combination of references would not result in Applicants' claimed invention.

For at least the reasons above, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 8, 15, and 23 include limitations similar to those of claim 1 and were rejected for the same reasons as claim 1. Therefore, the arguments presented above apply with equal force to these claims, as well.

Regarding claims 2, 3, 9, 10, 19, and 20, contrary to the Examiner's assertion, the cited references fail to teach or suggest that the prefetch unit is configured to prefetch a line into the instruction cache comprising instructions that corresponds to operations that precede and/or follow a branch in the evicted trace. The Examiner previously cited Jourdan (in column 2, lines 40-46) as teaching these limitations. This passage states, in its entirety, "The execution core reports the branch miss prediction back to the instruction supply engine in the form of a command such as "jump-execution-clear" (jeclear). The instruction supply engine must then retrieve instructions from the cache system in order to keep the execution core as busy as possible, and avoid "bubbles" in the instruction pipeline." **Applicants have argued that this passage has absolutely nothing to do with prefetching lines of instructions that correspond to operations that precede or follow a branch in an evicted trace.** For example, it describes the operation of the instruction supply engine to retrieve instructions from the cache system (not fetching or prefetching them into the cache system) in response to a branch misprediction, which has nothing to do with an evicted trace.

In the Final Action, the Examiner submits, "Since these instructions are located in a block, the remaining instructions inherently correspond to operations that either preceded or follow an evicted branch instruction." The Examiner seems to be referring to his additional citations in Darcy, although he does not cite anything in Darcy in reference to claims 2, 3, 9, 10, 19, and 20. However, as noted above, Darcy teaches a higher level cache pushing a block of instructions to a lower level cache, not prefetching a line of instructions from system memory in response to a trace cache eviction. Therefore, the additional limitations related to prefetching recited in claims 2, 3, 9, 10, 19, and 20 are clearly not taught or suggested by Darcy. Furthermore, in Darcy, there is no mention of demoting additional instructions (i.e., instructions that precede or follow a demoted block, whether it corresponds to an evicted trace or not) as part of the demotion

mechanism. It is not clear what the Examiner's remarks regarding "remaining instructions" signify, since Darcy teaches demoting an entire block of instructions into a lower level cache. **In light of the discussion above, Applicants assert that the Examiner's conclusions are completely unsupported in the references themselves.**

For at least the reasons above, the rejection of claims 2, 3, 9, 10, 19, and 20 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claims 4, 11, and 21, contrary to the Examiner's assertion, the cited art fails to teach or suggest, for example, *wherein the prefetch unit is configured to prefetch a plurality of lines of instructions into the instruction cache in response to the trace being evicted from the trace cache*. In the previous Office Action and in the Final Action, the Examiner merely states, "This limitation appears to be already included in the independent claims." Applicants again assert that the arguments presented above regarding the independent claims apply with equal force to these claims, as well. However, the Examiner is mistaken in his assertion that this limitation is included in those claims. The independent claims recite, for example, *wherein the prefetch unit is further configured to prefetch a line of instructions into the instruction cache in response to a trace being evicted from the trace cache*, i.e., a single line of instructions, not **a plurality** of lines of instructions. Applicants assert that this limitation is also not taught or suggested by Jourdan, Darcy, or the combination thereof.

For at least the reasons above, the rejection of claims 4, 11, and 21 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claims 5, 12, and 22, contrary to the Examiner's assertion, the cited art fails to teach or suggest, for example, *wherein the prefetch unit is configured to prefetch a number of lines that is proportional to the number of branch operations comprised in the evicted trace*. The Examiner cites Darcy (in column 30, lines 6-25) as teaching this limitation, submitting, "Indeed they will be the same number instructions which are clearly proportional." The Examiner seems to be referring to the fact that the demoted

block has the same number of instruction before and after demotion to a lower level cache. **The Examiner's remarks are not applicable in teaching the limitations recited in Applicants' claims.** For example, claim 5 does not recite prefetching a number of instructions that is proportional to a number of instructions in an evicted trace, but recites prefetching a number of lines of instructions (i.e., where each line includes multiple instructions) that is proportional to the number of branch operations in an evicted trace (i.e., not the total number of instructions in the trace). **Furthermore, as discussed in detail above, the combination of Jourdan and Darcy does not teach the prefetching of Applicants' claims, with or without this limitation.**

For at least the reasons above, the rejection of claims 5, 12, and 22 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claims 6 and 13, contrary to the Examiner's assertion, the cited art fails to teach or suggest, for example, *wherein the prefetch unit is configured to inhibit the prefetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the line of instructions is already stored in the instruction cache.*

The Examiner has taken Official Notice that it would have been obvious at the time of the invention for one of ordinary skill in the art to take the invention of Jourdan and utilize a cache system that, when presented with information to store, checks if that information already exists and inhibits duplicate storage. The Examiner submits that within the invention of Jourdan, the combination would check when there is an eviction from the trace cache. The Examiner submits that this technique is extremely common practice within cache systems and that storing the same information within different portions of a cache is a waste of resources that can create a detriment to the processing system with regards to space, cost, power and speed. Applicants have repeatedly traversed the Examiner's taking of Official Notice with respect to the teachings, or lack thereof, in Jourdan. Applicants have argued that while some cache systems may include a feature that prohibits storage of duplicate information, there is nothing in any art of

record describing that such a feature is inherent or well known in the particular cache memory system of Jourdan, or that the lack of such a feature would be detrimental to the cache memory of Jourdan in any of the ways listed by the Examiner. Applicants again assert that some cache memory systems have perfectly valid, performance-related reasons for allowing, or even encouraging, the storage of some duplicate information in an instruction cache and/or in a trace cache. For example, in some systems, duplication of at least a portion of some traces in a trace cache may allow for better performance by allowing both the predicted-taken and predicted-not-taken targets of each branch to be included in a trace cache. Other systems may allow or encourage duplicate information for other reasons. **Applicants assert that the above-referenced feature of claim 6 is not well known in the type of cache system described by Jourdan.**

In the Response to Arguments section of the Final Action, the Examiner states. "Examiner asserts that most every cache system follows a general principal: if a cache hit occurs (a case where a duplicate would exist), data is read from the cache and no write is made; if a cache miss occurs (a case where no duplicate would exist) data is written and no read is made from the cache. Consider, for instance, Fujimoto (U.S. Patent No. 6,145,055) col 1 line 30-57)." Applicants first note that both the Examiner's remarks and the cited passage of Fujimoto describe the actions taken in a cache system in response to a cache hit or a cache miss. Fujimoto describes that if a cache miss occurs "one data stored in the cache is expelled and replace action for storing new data is performed." In other words, this passage describes that in response to a cache miss, new data that is currently needed is fetched and this data replaces the expelled data in the cache. Alternatively, in the case of a cache hit, there is no eviction from the cache. Therefore, this case has nothing to do with the limitation under discussion. **Applicants again assert that these descriptions of the actions of a generic cache system in response to cache hits and misses has absolutely nothing to do with prefetching actions that may or may not be taken in response to an eviction from the cache, much less in response to the eviction of a trace from a trace cache, as in Applicants' claim.** Fujimoto does not describe any cases in which instructions are (or are not) prefetched, dependent on data evicted from a cache or whether this data is duplicated in another level of cache.

Furthermore, the cache system of Fujimoto does not even include a trace cache. **Therefore, the fact that the cache system of Fujimoto does or does not duplicate information in a given cache level in response to a cache hit/miss implies absolutely nothing about whether the cache system of Jourdan, which includes a trace cache, would or would not prefetch a given line of instructions into an instruction cache in response to an eviction from the trace cache.**

Further regarding claims 6 and 13, as noted in Applicants' previous Response, these claims do not merely recite that a cache memory is presented with duplicate information that it determines not to store, but instead recite that *the prefetch unit is configured to inhibit the prefetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache if the line of instructions is already stored in the instruction cache*. In other words, in Applicants' claimed invention, duplicate information is not presented to the instruction cache (i.e., after being fetched or prefetched) and checked to see that it duplicates information already present. Instead, the line of instructions (which would not be duplicate is not prefetched at all. **Applicants again assert that this is clearly not taught by the cited art.**

Finally, the cited art does not teach or suggest anything about ***inhibiting*** *the prefetch of a line of instructions into the instruction cache in response to the eviction of certain traces from the trace cache*, as recited in claims 6 and 13, nor has the Examiner included anything in his remarks explaining how he believes the references teach this limitation. **The Examiner's statement, "within the invention of Jourdan, the combination would check when there is an eviction from the trace cache" is completely unsupported by the cited art, as Jourdan does not describe prefetching of instructions into an instruction cache in response to an eviction from the trace cache at all.** Therefore, there would be no reason to inhibit such prefetching.

For at least the reasons above, the rejection of claims 6 and 13 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 16 and 17 include limitations similar to those of claims 6 and 13 and were rejected for similar reasons. Therefore, the arguments presented above apply with equal force to these claims, as well.

Allowable Subject Matter:

Claims 7, 14 and 18 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form. Applicants respectfully thank the Examiner for consideration of these claims. However, for at least the reasons above, Applicants believe that the claims are allowable as currently written. Therefore, no amendments have been made to these claims in the present Response.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91600/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: December 18, 2008